Assignment For Day 2 😊🥇🌱

1. **What is lexical structure?**

=> Lexical structure of a language is lowest-level syntax of a language. Lexical structure defines the rules of how a variable name should look like what are the delimiter characters for comments and how one statement is separated from the next.

2. **What is Unicode?**

=> Unicode is an international encoding standard by which different characters, symbols and numbers are assigned a unique value that can be used across different scripts and languages.

3. **Explain all the keywords present in the JavaScript with examples.**

=> await: The await character is used to wait for a promise.

break: break statement breaks out of a switch or a loop.

case: case statement is used to define different conditions/cases in a switch block.

catch: the catch statement is used to define a block of instructions for what should happen when an error occurs in the try block.

class: the class keyword is used to create/define classes in JavaScript.

const: const keyword is used to define constant/ values that are not going to change throughout the lifecycle of a program.

continue: continue statement breaks one iteration when some specific condition is met and continue with the next iteration.

debugger: the debugger statement stops the execution of JavaScript and calls the debugging function if any, and if debugging is not available it has no effect.

default: the default keyword is used to define the instruction which should happen when no case is met in a switch block.

delete: delete keyword deletes a property from the object.

do: in a do-while loop do block specifies the instructions that is needed to be iterated.

else: else block specifies the instruction which should get executed when no condition(s) met in an if/ (if-else) conditional blocks.

enum: enum describes a special class which is made up of const or unchangeable variables.

export: export lets you export the module from one file to be available in other files.

extends: extends keyword is used to specify the parent class of a child or derived class.

false: Boolean false value.

finally: statements in finally block are executed regardless of if there was an exception in the try block or not.

for: for specifies the set of instructions that are needed to be met for the following statements to be iterated upon.

function: function keyword lets you describe your functions/ methods.

if: if holds the value for which if true the if block executes.

implements: *FutureReservedWord.*

import: import lets you import exported modules from other files.

in: returns true if a specifies property is in the object

instanceOf: is used to check the type of an object at the run-time.

interface: interface is a *FutureReservedWord* that is at this point of time it does absolutely nothing.

let: is used to define variables

new: new is used to create an instance of the object which has a constructor.

null: describes no value.

package: *FutureReservedWord.*

private: *FutureReservedWord.*

protected: *FutureReservedWord.*

public: *FutureReservedWord.*

return: returns output of a function.

super: is used to call the constructor of a parent class.

switch: is used to describe a switch conditional block

static: *FutureReservedWord*

this: this refers to an object.

throw: is used to throw a user-defined exception

try: try block lets you specify the piece of code that is tested for error while it is being executed.

true: Boolean trye value.

typeOf: will return the type-of variable when called.

var: used to declare variables in JavaScript

void: evaluates the given expression and then returns undefined.

while: while specifies the condition which when met the specified code that needs to be iterated upon is executed.

with: with statement extends the scope chain for a statement.

yield: *FutureReservedWord*

4. **What are shorthand operators, explain with a suitable example?**

=> Shorthand expressions are a shorter way of describing something that is already present in the language.

Ex:

```
x += 5        =>       x = x + 5
```

5. **What is “use Strict” in JavaScript?**

=> “use strict” declaration throws an error when a variable is not described beforehand.

Ex:

```
function myfunc() {
car = ‘Toyota’ //will not throw an error
}
```

```
function myfunc() {
“use strict”
car = ‘Toyota’ //will throw an error since car is not described.
}
```